91802
new 82-10

LEGAL NOTICE

This report was prepared by Ontario Hydro and, while effort has been made to assure the accuracy and applicability of the material contained herein, neither Ontario Hydro nor any person acting on its behalf,

(a) makes any warranty, express or implied with respect to the use of any information, apparatus, method or process disclosed in this report or that such use may not infringe privately owned rights; or

(b) assumes any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method or process disclosed in this report.

ontario hydro
research division

COMPLIMENTARY COPY
Please Credit Ontario Hydro Research if Reproduced in Whole or in Part

TO: Dr. D. Mills
Director of Research

ROBOT NAVIGATION WORKSHOP AT IJCAI-89

S.B. Nickerson

The International Joint Conference on Artificial Intelligence (IJCAI-89) in Detroit, August 20-25, 1989, was attended. Because of our PRECARN project in "Active Vision Navigation of a Mobile Robot in a Known Environment", the highlight of the conference was a panel discussion on "Robot Navigation". The panel chairman was David Miller (Jet Propulsion Laboratory) and the panelists were:

- Rodney Brooks (MIT)
- Raja Chatila (LAAS-CNRS, France)
- Scott Harmon (Robot Intelligence International)
- Stan Rosenschein (Teleos Research)
- Chuck Thorpe (CMU)
- Chuck Weisbin (ORNL)

Submitted material by the chairman and panelists (D.P. Miller et al, "Robot Navigation" in Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, 1672, 1989) is attached as Appendix A. It provides a good summary of the discussions, a transcript of which is attached as Appendix B.

Approved by:

A.F. Baljet
Manager
Electrical Research Department

Submitted:

S.B. Nickerson
Theoretical Physicist
Science Section

job	file	date	report no.
16-554	810.1465/X803.23	December 6, 1989	89-263-K

APPENDIX A

"PROCEEDINGS OF THE ELEVENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE (IJCAI-89)," DETROIT, AUGUST 20-25, 1989

Robot Navigation

Panel Chairperson:
David P. Miller
Jet Propulsion Laboratory/California Institute of Technology
Pasadena, CA 91109 U.S.A.

Panel Members:

Rod Brooks,
AI Lab, MIT
545 Technology Square
Cambridge, MA 02139

Stan Rosenschein
Teleos Research
576 Middlefield Rd
Palo Alto, CA 94301

Raja Chatila
LAAS-CNRS
7, Ave. Du Colonel Roche
31077 Toulouse Cedex, France

Chuck Thorpe,
The Robotics Institute, CMU
Pittsburgh, PA 15213

Scott Harmon,
Robot Intelligence International
PO Box 7890
San Diego, CA 92107

Chuck Weisbin
Oak Ridge National Lab
PO Box 20008, Bldg 6025, M.S. 6364
Oak Ridge, TN 37831

Abstract

Robot navigation is a problem that encompasses most of the major areas of AI research. Machine architecture, search, knowledge acquisition and representation, planning, scheduling, reaction, perception, and of course robotics, all can play integral roles in a mobile robot navigation system. For this reason, a large part of the AI community is now interested (and has been since STRIPS was used to guide the Shakey robot) in mobile robot navigation.

1. David P. Miller

There are two pieces of AI folk wisdom that show why mobile robot navigation is an important problem for researchers to address: 1) *Simulations are doomed to succeed* 2) *Real robots don't work; at least not the way you thought they were going to*. These bits of knowledge point out the reasons why robotics is becoming more important to AI and why mobile robot navigation is becoming one of *the* problems in this area.

For years researchers in AI have worked on narrow problems in the field, and many promising solutions to these problems have been found. Work in spatial reasoning, resource allocation, task planning, sensing, sensor fusion, etc. have resulted in a variety of very interesting work. The question arises: why, with all this previous work, are there so few functioning systems that make use of these modules? I believe the answer, at least in part, is that the simplifying assumptions made about the modules associated with the module actually being designed are too simple. Implementing a module in a real robot system makes it difficult to make oversimplifications.

Robot navigation covers most of the major areas of AI research. Issues, for example, in representation that used to be discussed in AI circles in terms of abstraction, elegance, and clarity, are discussed in mobile robot circles in terms of whether or not it works.

This panel is composed of people who have made systems that work. Interestingly there still remain unresolved issues. Some of those that are addressed below are:

° What aspects of the robot's environment need to be modeled internally by the robot?

° What kind of sensory input is appropriate/vital for navigation in natural terrains?

° Should different sensor modalities be fused or diffused?

° How much execution monitoring should be preplanned?

° Is general purpose route planning any easier a target than general purpose planning?

2. Rodney A. Brooks

The field of robot navigation has in many ways become abstracted away from reality and many groups and individual researchers have spent large amounts of time solving problems which I believe to be irrelevant to real world mobile robot navigation. Actual experiments with mobile robots, such as the ALV, have shown that good performance can be achieved by using an approach which is quite different from the traditional knowledge-based AI approach. In contrast, such traditional approaches have not shown anywhere near as impressive performance in the few cases that they have been implemented on real robots.

In particular I will argue that

o Explicit complete three dimensional world models are unnecessary. Rather, only task dependent aspects of

the world need be modeled, such as clear paths ahead, dynamic obstacles which could cause navigation problems, and landmarks need be explicitly extracted from the sensors.

o Different modalities from the sensorium do not necessarily need to be fused. Rather they may be fed directly to appropriate task achieving behaviors as appropriate. For instance laser scanning data might be used to correlate with an a priori map in order to do localization if that happens to be important for some particular task, whereas sonar data might be used only for local obstacle detection and never get correlated with the map.

o Execution monitoring should not be preplanned. It should be built.

o The appropriate sensorium is not dependent just on the fact that the robot is to navigate over natural terrain. Rather it is critically dependent on both the reason for needing to navigate, on the relative scale of the vehicle and the features over which it must travel, and on the method of locomotion of the vehicle. For instance, a multi-legged machine that is to crawl over a terrain consisting of densely packed boulders of the same size, and whose goal is to move in some general direction on an exploratory mission, might work well with only force feedback on its legs, feelers (or whiskers) to predict obstacles just ahead, and a compass or sun sensor for general direction.

o General purpose route planning is a red herring. What is needed is not a route plan, but an ability to get to where you want to go. Route planning is an unnatural abstraction of this task, and the Hughes ALV work has clearly demonstrated the pitfalls of this abstraction.

3. Raja Chatila

Consider navigation (reaching a given goal) in a gradually discovered environment, with various conditions for perception and motion (lighting, kind of terrain, shape and behavior of objects: fixed, movable, moving, etc.). The robot has specific intrinsic constraints (sensors, computing capacities, physical features, locomotion structure), and a task to perform.

1. Representations of the geometry, topology and some semantics of the environment are necessary. Exact model contents are sensor, robot and task dependent. Some tasks may require no model, but one is needed in order to:

a. self-locate with respect to the environment. If this is impossible, reaching a given goal (that is not in sight) will be impossible in general.

b. plan and execute a collision-free path to reach a given goal minimizing a criterion (energy, distance, time,...) and satisfying some constraints.

2. Natural terrain is essentially unstructured. Thus navigation requires data on the terrain and its nature. A depth map (e.g. provided by a 3D laser range-finder) fused with color vision is very suitable for extracting geometric features with attributes on navigability. Dead-reckoning and sensors on the locomotion structure might be vital for detecting slippage, unstable postures, etc. Sensors providing a poor but quick response (e.g. sonar) might be useful for monitoring.

3. Each type of sensor has limitations and uncertainties. Since the environment and robot state change, a single sensor will not be sufficient in general to build consistent environment models. Therefore fusion (spatial and temporal) of sensory data is necessary for navigation but not systematically, for example, for obstacle avoidance, tracking, etc.

4. Execution monitoring ensures correct task execution, reactivity to events and error recovery. Reflex loops must exist at several levels, but provision should be made to make them context and task dependent. This implies that decisional capacities, for analyzing a situation and foreseeing some events to be monitored, are necessary. Thus, execution monitoring should be preplanned, with some flexibility given to the on-line system.

5. Path planning itself is complex in general, and cannot be solved by local methods or by "trial and error". But navigation is not only geometric reasoning since the robot has a task to perform that may put various constraints on it. Feasible trajectories should be planned, using a topological and semantic knowledge on the environment, according to the capabilities of the robot's locomotion system. The problem is thus of the same nature as task planning in the general case.

4. Scott Harmon

As a general rule, a robot's configuration depends entirely upon its task. Its perception, modeling, execution monitoring and planning capabilities are driven completely by its task goals and environment. Absolutely no more capability than is needed should be incorporated.

Following this philosophy, a robot should model only as much of the environment as it needs to reliably accomplish its task. This depends upon both the specifics of the robot and the task. If the robot is relatively insensitive to the environment's complexity then little of the environment must be modeled.

A robot's sensory input depends completely upon the environment's character and the task demands. Changes in any of these factors alters the sensor information requirements. The robot should sense the environment's geometrical nature at a scale which is significant to its mobility mechanisms. Any environmental elements which affect robot mobility must be sensed in order to guarantee reliable passage. The robot may need to characterize significant nongeometric conditions as well.

Sensor fusion is primarily useful for obtaining more information about a situation than is available from independent sensors. If independent sensor measurements provide all the information at the accuracy and certainty which is adequate for planning and successful execution then no sensor fusion is needed. However, this is seldom the case for complex mobility tasks because of environmental complexities and uncertainties. Sensor fusion enriches the understanding of the task state in geometric space, in time, in modality and in certainty.

All execution monitoring must be preplanned. Effective execution monitoring is simply impossible if a priori expectations of execution results are not available. However, many other conditions must also be met to guarantee monitoring effectiveness. The important components of the execution process must be sensed. Execution monitoring is extremely useful if the execution process is nondeterministic. The particular execution process must be important to the success of

accomplishing the task. If the process is not very important then don't monitor it; if it is very important then monitor it closely. Finally, the robot must be able to take some corrective action if the execution process goes awry. Monitoring is only useful if something can be done when a problem is detected. Monitoring becomes more detailed as more corrective actions are available.

Finally, truly general purpose route planning for a complex robot in a complicated environment reduces to general purpose planning so they, in the limit, have the same level of difficulty.

5. Stanley J. Rosenschein

The navigation task, like other underspecified tasks, can be made complex enough to require modeling of virtually any fact about the environment and detection of almost any imaginable condition. Navigation means traversing a continuous path that terminates where the robot needs to be. Because the reasons for taking one path rather than another (e.g., goals of seeing or being seen, not falling into elephant traps, avoiding poison ivy, etc.) are so diverse, and because conditions to be maintained along the way may be qualitative and not merely positional (e.g., maintain ability to see the road ahead), the primary research challenge is in integrating navigation and other perceptual and reasoning processes. Practical, general-purpose navigation appears to be as hard as general-purpose planning.

On the question of sensor fusion, although some have argued for autonomous, task-specific sensing modalities, we feel that, whenever practical, sensor modalities should be fused, for two fairly obvious reasons: (1) so that more conditions can be recognized, making possible more sophisticated robot control and more fluent transfer between tasks, and (2) so that the same conditions can be recognized more robustly through redundant evidence.

6. Chuck Thorpe

1. What aspects need to be modeled?

Answer: As many as possible, and as explicitly as possible. There are always models of the environment, even if no more than the distance to the nearest sensed object. Robots without models cannot accomplish interesting tasks, at least not efficiently. It is possible to build an extremely simple robot with no models, only reflexes, that does some task such as wanders a table top and turns when it encounters an edge. Such robots don't need models, computers, or even electronics; everything is done mechanically. But a robot to follow a road, or clean the floors, or perform any other really useful task, must have a map of its environment. Any intelligent coupling of perception to action, any planning at a higher level than a reflex, needs a model of the world.

In short, robots without models may make great protozoans, or even with much cleverness may produce artificial insects; but I wouldn't trust one to be a chauffeur.

2. What kind of sensors is needed for natural terrains?

Answer: Maybe not as much as we have thought. For our AMBLER project), the primary sensors will be range data and foot forces. In an environment such as Mars, the entire problem is geometry and on surface strength.

I think a much richer set of sensing (sensors and processing) may be needed for travel in semi-natural terrains, where it is important to recognize flowers and roads, and to properly interpret stop signs and moving cars and pedestrians and police cars and ...

3. Fusion or fission?

Answer: Both, of course. There is interesting evidence of two different biological perception mechanisms, one for locomotion and one for recognition. In practical robot vision systems, there may be several different activities going on in parallel. Each of these activities may need information from several sensors, such as Anita Flynn's work on fusing sonar and IR range data to build maps. So at least some of the data needs to be spread over at least two different processes.

4. Preplan execution monitoring?

Answer: Definite maybe. I greatly prefer to keep my planners so simple that they can be rerun at almost every step, or at least any time something goes wrong with the original plan. So rather than monitor execution, I prefer to update the world map / model, and replan. In the robots I've known and loved, perception has always been much slower than planning (and much less reliable), so the extra planning wins, especially if it can produce a plan which does less sensing.

5. Route planning vs. planning.

Answer: Route planning is much easier, except in really contrived cases. It is of course possible to fold any problem into route planning, e.g. "plan the route that would be necessary to bake a cake" could incorporate all the steps in cake-baking-planning. It is of course also obvious that optimal route planning can be tough, e.g. traveling salesman. But practical path planners, looking for "sufficing" rather than optimal paths, can be much simpler than full-blown planners.

7. C.R. Weisbin

1.The aspects of the robot's environment which need to be modeled seem to me to depend critically on the task (e.g., navigation in a known world, exploration and goal finding, ability to deal with unexpected events, etc.). We would provide an "a priori" model appropriate for the task domain, and confirm that model dynamically based upon sensing feedback.

2. Sensory input for navigation in natural terrain seems again to depend on the task and the available processing time. Sensors for autonomous navigation of an automobile moving at 60mph, could be different than sensors required for delicately defusing a bomb.

3. If by fused or diffused, you mean that independent sensor information is consistently combined (fusion) versus independent sensor data used for different task functions, we would tend to support multisensory fusion where time and resources were available, and independent sensor readings for different task functions where speed or cost effectiveness dominated.

4. I do not understand how execution monitoring can be anything but preplanned. Do you not need to anticipate the types of events to monitor before implementation of monitoring?

5. General purpose route planning can be envisaged and extended to cover many, if not all, of the most general purpose planning paradigms; but I do not think this question (as it is posed) is very important.

APPENDIX B

11TH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE

.....Welcome to this panel session recorded live at the 11th International Joint Conference on Artificial Intelligence. On this cassette you will hear robot navigation. Now to our program.

MILLER

Now, the reason why we're having this panel is because robot navigation seems to tie in a lot of different aspects of traditional AI problems - sensing, perception, reasoning about actions, planning, deciding what the robot actually believes or doesn't believe, what it knows. Most of the different sub-fields of AI can be addressed in one way or another through this subject and the interesting thing about this subject is that it forces you to be a little more realistic than most other AI problems in that you actually have a piece of hardware (hopefully) that you can see will either work or not work and if you're trying to sort of simplify the problem one way or another it becomes more obvious when people come and visit your lab and all the walls are painted black and they have white stripes along the edges or something like that. It's more obvious what your simplifying assumptions are rather than simply having a piece of barried code somewhere. So, for this panel, my name is David Miller from JPL. For this panel, we have a bunch of experts on robot navigation and robotics. Everyone on the panel has participated on a major robotics project one way or another and everybody has had experience with real robots. Just to introduce the panelists going from right to left is Raja Chatila from LAAS, Chuck Weisbin from Oak Ridge National Labs, Scott Harmon from Robotic Intelligence International, Rod Brooks from MIT, Chuck Thorpe from CMU, and Stan Rosenschein from Teleos Research. All of them have worked on robot projects and are going to spend a few minutes telling you either about their projects or about other people's projects and why they're not as good as their projects or some combination thereof. The format for the panel today is each speaker is going to give about a 10 minute presentation on either their work or these issues or some combination thereof. After that's concluded we're going to try and have a discussion on some of the issues and disagreements that have been brought up and after that we will just have an open discussion and open it up for questions. So for the first speaker, the order of speakers for today, Scott Harmon is going to start off and give the first presentation.

HARMON

I want to thank Dr. Miller for giving me the chance to pontificate on robot navigation - a subject that has interested me for a number of years. Being the first speaker is a little bit hard with a group so esteemed as this because you hardly know what they're going to say about you after I sit down so I've taken this opportunity to talk a little bit about some of the history of robot navigation and some observations that I've made over the past 12 years in which I've been involved. The first thing is that we have several areas of sensing - the ability to perceive the environment, mobility - the ability to get around and then reasoning and planning and some demonstrations that we've made. So here are some of the things that we've seen occur that weren't true say 12 years ago. First, we have cheap acoustic range finders available so all of you that are interested in doing that kind of range sensing - you can do that very inexpensively now and very robustly. Readily available vision processing, you can buy vision processes for your IBM PC or MacIntosh now which we couldn't before. We had to spend hundreds of thousands of dollars for the people working in stereo vision have made it much faster which makes it more realistic to put on a robot that's moving continuously through its environment. Laser range finders have evolved. They're still not very cheap right now. It would be nice if more of us could afford them but they're something that are available that weren't available say 10 years ago. We can do geometric modelling better. Some technology that has evolved out of this community and we know how to combine information better. Right now you can

go and buy platforms so you don't have to build a mobile robot, you can buy the platform and then instrument it the way you like or you can even buy it instrumented if you'd like and we've seen legged vehicles being demonstrated even out of doors and in a wide variety of environments. So that technology has come a long way. You can get knowledge-based tools for your IBM PC as well, widely available and very inexpensive and for small computers and we have today a number of well characterized geometric planners that you can use. So that's not a problem you necessarily need to deal with if you don't want to. In the demonstrations we've seen over the past 10 years high-speed road following - the Germans have demonstrated 60 kph or better. Low-speed road following, the US has demonstrated 30 kph. Obstacle avoidance and a number of different kinds of environments. Computer-aided remote driving has evolved quite well right now and we have a considerable amount of remote driving experience. You might ask, well what is left? A number of problems remain that we're approaching. In the hazard avoidance pathfinding area, I'd like to see a low-cost hazard detection sensor that you can get that will characterize the geometry of the environment well and that's not available and the type of hazard that I call non-geometric and another way of explaining that is it's what you see is not what you get. Geometric hazard would be you could look out and see a pole in front of you or a rock, a non-geometric hazard would just look like a traversible area but it's not and might have low load bearing strength or low coefficient of friction neither of which are very easy to detect and in fact the Germans, who have done high-speed road following, have found a particular problem with being able to do advance sensing of the coefficient of friction - something very hard to do but very important in controlling the vehicle as any of us know who have driven on rain or snow covered roads. Surprisingly enough we can't locate robots very well when in their environment after they've travelled a short distance so the whole area of position estimation remains unsolved. It would be nice to find a low-speed true ground speed sensor that you can get that's accurate. That's something that's not available today although there are a number of true ground speed sensors that use radar or ultrasonics or dopler effect. They don't work very well at low speeds and for those of us working in the environment, a lot of us don't go at much more than a kilometer per hour and these ground speed sensors work at a one mile per hour or above. It's very difficult to find out absolutely where you are once you've moved a certain distance. It's easy to figure out how far you've gone and if you're in the right environment where you are or which direction you've moved but knowing where you are in some absolute coordinate reference frame turns out to be a very difficult thing to do particularly if you're outdoors. You might resort to landmark navigation but that isn't a very well developed technology yet. It would be nice to have a non-magnetic heading sensor. Gyros for instance, drift very rapidly and most of the ways that were used to figuring out which direction the robot is moving in in say an earth-reference frame is by the magnetic field. But if you go to Mars there isn't one, but it would be nice to have, especially for that project, a non-magnetic heading sensor. In addition, magnetic sensors are very sensitive to the local environment so if you haven't carefully planned that, haven't carefully identified all the magnetic anomallies you might find your robot driving in circles. We don't do very well in controlling vehicles that don't pivot on their own axis and so as a result you can shrink them to a point in configuration space. Non-holonomic vehicles are those that require moving to turn and as a result of that they require a lot of extra planning to path-plan their paths. We don't do very well in figuring out how vehicles interact with the ground. So if you're moving over anything but a flat surface, it's very difficult to actually control the vehicle to, make it go which direction you want and more and more complex vehicles are evolving and to be honest we're losing the ability to control these vehicles very well and you have to consider that in something say like a Mars rover where you have three bodies and all the roll and pitch between them and that all the wheels are contacting the round, you effectively have a 12 of freedom system that is coupled for the ground so it's very difficult to solve the dynamic equations through these vehicles and they take a long time. So we need lots of work in these areas. And the experience - we would all like to be able to advocate that we can do autonomous navigation but we haven't done it very much in spite of the fact that Rod's built 10 robots and Chuck has 5 or 6 and the rest of us probably another 5 or 6 between us. That's not very much considering for instance the manipulator field where hundreds and hundreds of manipulators have been built and exist in almost every university. We really need a lot more experience in both operating semi-autonomous vehicles just in simple environments outdoors as well a in rough terrain and we need more experience in being able to deal with non-geometric hazards. These things that

you can't necessarily discern the hazard from a vision system for instance and then the whole area of unknown terrain is open. So in my last two minutes I've summarized the arguments of each of the people here in, again another pleasure that I have, of being the first speaker and I've used something that's near and dear to my heart. I'm a restaurant fanatic so I have the restaurant index for how these people have answered the different questions. Whether a robot should have an internal model? How much sensing? How you do the data handling from that sensing and the execution monitoring? And I've characterized them this way, Rod Brooks for instance is very austere in internal models so I call him the salad bar type and Raja is very discerning, comes from a land of great restaurants and this restaurant is recommended to me as being one of the best in France. I, on the other hand, am inbetween Rod and Raja in terms of being discerning. Both Stan and Chuck Thorpe believe that you should do lots of internal modelling and I call that the Picadilly Cafeteria model whereas Chuck Weisbin is somewhere in between Raja and me. In terms of sensors, it's like what drink are you going to order, well Rod would order water. Both Raja and I believe that you should choose the wine to depend on the meal which means you choose the sensor to depend on the task. Stan didn't give me a vote to launch a harpoon at him. Chuck Thorpe, house wine man and Chuck Weisbin brought a very interesting thing in. He said that the number of sensors that you choose and the type that you choose depends not only on the task but also on the bill or how much computing you've got which is a new thing. In terms of data handling, Dave mentioned the fision-fusion I've just summarized. Some people believe in both fis and just fis in just fuse or fis and fuse. Execution monitoring is like making a reservation at a restaurant. Rod would just show up. Three of us would reserve ahead because we believe in planning execution monitoring ahead and Chuck he'd just call ahead to make sure that there was space. Thanks very much.

CHATILA

I will start by introducing myself a little bit. I work in France in the city of Toulouse on mobile robot project called Hilare which means Hilary. Ten years ago at IJCAI '79 in Tokyo we presented the first paper on this work and now may be it's time, 10 years after, to summarize some lessons that we learned from this experience and I will articulate my presentation around the questions that were asked by Dave and that he presented in the beginning. So, the first question is, what aspects of the robot's environment should be modelled internally? Well, this is not the ??????????? you knew but... I think that the robot is to perform tasks more or less complex in a three-dimensional environment. Therefore, it needs first a sense of the geometry of this environment - geometry represents position, shape of objects, that it will interact with and if it is not able to have this sense of geometry, it will not be able to accomplish a given task in general. Of course, with no model it will always be able to may be avoid an obstacle, may be stupidly avoid it but it will not be able to recognize anything about it. Topology is a very important item concerning navigation. Topology gives you relationships between objects, between objects and free space and free space is the central issue in navigation and usually topology is presented by some discretization of free space. Semantics is an important thing also. We used to think about robot navigation as, for example, obstacle avoidance but sometimes when you have a completely obstructed path, may be the obstacles are movable and the robots should move them in order to go through. So some semantics on the objects are also necessary if you consider navigation task as a general task. This means that you have also to be both quantitative and qualitative concerning the models. Shape might be qualitative and position is quantitative in general. In order to represent things internally, you don't just build models from scratch. They are task-dependent and of course sensor dependent. This means that usually you have to provide the robot with some generic primitives that will be instance-shaded according to what is perceived. Second, question was on natural terrain. Well, natural terrain usually or historically may be - most of us worked on indoor robots and in the past few years natural terrain has become a more and more central issue especially, for example, when the MARS project now or on the ALV project before. What characterizes natural terrain is that at the locomotion and perception system scales, of course, this terrain is not structured. It doesn't have a priori regularity neither a priori landmarks and the objects are in general also regular and had more or less complex shapes. I believe that in this case the sensory inputs must be a fusion of many sensors as I will mention this afterwards and we need at least three kinds of sensory inputs. One for terrain presentation and robot

localization with respect to the environment. This I believe will be based on three-D vision plus possibly colour which is a very important point and range data mainly for example laser range finders. Of course, in some cases range data will not be very good for example when you have bushes and very textured situations. For motion control that is actually the control of the movement of the robot, you need initial devices and you also need sensor specific system and Scott just mentioned a few of them and you need also some sensors or some sensor modalities for dynamic processes. For example, for obstacle avoidance if you have to quickly detect and avoid an obstacle that is possibly moving, this means that you have to use modalities with quick response time in this case. Third question was, should we use fusion or not for sensory data? Well, I think it's quite obvious that a given sensor has limitation in terms of accuracy, resolution. It has its own ambiguities, sensitivity to the environment, for example, try to use a camera in a completely obscure room and of course the given sensor should be adapted to the task. Therefore, I believe that, in general of course, this shouldn't be systematic. In some cases, you don't need to fuse sensors but in general it is a plus and sensor fusion will be in two aspects and especially in the temporal fusion that means when you have to fuse several views. This is an essential point in robot navigation because the robot should incrementally build its map. Some basic techniques that I won't develop now are known for fusion - mainly filtering for quantitative fusion. Fourth question was execution monitoring. Well, this is the central point. What is execution monitoring? Execution monitoring of a task ensures three aspects. First, correct task execution. Second, reactivity to events. Synchronous events that may occur during the execution of the task and third, error recovery. Well, I think that in order to be sure that you can have some reactivity you have to be prepared for it. Therefore, as an example, you have to set you as the robot and the robot according to the task has to instance-shade or program itself some conditions, actions ????????????? that detect possible events and react in the reflex reaction. Why does this has to be programmable and not built-in or wired because we want a controlled reactivity. We don't want systematically the same reflexes whatever is the task, whatever is the context. Error recovery: I don't believe that re-planning complete re-planning is a good thing. Instead, local corrective actions should be made first. This to ensure a gradual response to asynchronous events. The problem is that if you re-plan at every unexpected or asynchronous events you will lose a lot of time and local corrective actions will in general be sufficient and if this fails you will be able to re-plan in any case. Final question, route planning versus task planning. Well, if you consider this in its pure geometrical aspect, for example, when you consider obstacle avoidance in very cluttered space using configuration space approach and so on, in this pure geometrical aspect, the problem is well studied and its complexity is known. It's polynomial in their representation parameters, of course, if you have higher degrees of freedom you have an exponential growth with the degrees of freedom. If you have a discretization of the space you have also an exponential evolution with this discretization but for a given discretization and a given number of degrees of freedom the problem is polynomial. It is complex but it is polynomial. However, I don't believe that navigation is only a geometrical problem. Navigation is a task by itself and it's a task that is done in order to achieve another more general task which is related to general task planning but since the general tasks put constraints on the specific movement you have to go in this and this direction or to this sub-goals. You have to achieve some things while you are moving, therefore, these put new constraints and make the general purpose route planning equivalent to task planning. Thank you.

ROSENSCHEIN

I want to talk today about a few issues that arise in robot navigation and over the last few years, of course, we've seen a large number of robot projects that have a navigational component in certainly a lot of application areas that seem to require navigational ability of robots. Various industrial applications, transporting materials from one place to another, autonomous land vehicles, planetary rovers, etc. One is immediately led to ask the question why is navigation hard? Is this really a hard problem? In terms of the approaches we're probably going to hear about today, I'd say that the approach that I've advocated that's closest to Rod's in many ways and I'll try to argue for that a little bit. I'll let Rod do all the hard arguing later. I'll have a few comments at the end about some possible differences between my methodology and Rod's. Well, to answer the question why is robot

navigation hard, let's back up a little bit and ask the question why is robot perception and action hard? Since, after all, navigation is just a certain slice through the space of possible robot perception-action problems. One way you might model the design problem for a robotic system is just to conceive of the system as being controlled by a controller of some sort that's interacting with an environment and a continuous stream of input and output. The robot controller has certain capabilities. Here I've seen to imply that these capabilities are realized as separable modules, perhaps software modules and I'll argue in a second that that doesn't necessarily have to be the case. They might be more mushed together but basically the robot controller has to be able to extract information from the environment constructing something like an information state whether you think that information state is encoded in some explicit knowledge representation language or whether it's just encoded in a bunch of signals that are being sent around the circuit. Still, there is some information there and the designer has to have some conception of what that information is and how it's going to be used by the action selection component to generate actions which control the environment in the appropriate way. So why is this a hard problem? Well, the reason is because there are many, many sources of complexity lurking around that a designer has to grapple that are very challenging. There are difficulties in modelling the environment itself. The environment after all is a big place. We all live in it every day and it's not clear exactly what aspects of the environment need to be modelled for a given task using what abstractions and so on. There are always going to be unmodelled aspects of the environment so we're never going to have a compietely comprehensive model and yet to support a certain kind of behavior we're going to have to - when I say we I mean the designer not necessarily the robot - but somebody has to get a grip on what is going on in the environment. Similarly, there are difficulties in modelling the information states of the agent. In many cases these models are quite informal and they're just based on the designer's intuitive understanding of what the various internal states of the agent actually encode about the environment. In some cases, as in our own work, we try to be a little bit more rigorous about the mathematics of that kind of information encoding. In other words asking the question what does it mean for the states of an agent to encode information about the environment? What formal languages can we use to describe those encodings? How can we decouple the complexity of keeping those encodings up-to-date from the complexity of logical languages that we used to reason about them? Finally, there is the difficulty in specifying the task itself, I'd say the task of the robot it's supposed to be carried out. There are various subtleties there and formal languages have been invented to make subtle distinctions between various kinds of behaviour the robot could be carrying out. For example, the difference between achieving A and B versus achieving A and achieving B. They sound almost alike in English but when you get into the mathematics of it, they're quite different things. Now, given all that as background one might think well, robot navigation is a special case, the information states in this instance encode locational information. The actions affect location and orientation so we could imagine a perception component for example, that constructs for us some geometric description of the world and then gets mapped by the action component into the appropriate navigational actions to correct our course and get us to where we're going. This seemed to be what Scott was calling for when he said we need to have more accurate positional information. Speaking for myself and possibly for Rod, in many instances the information available to the robot is actually quite weak. Much weaker than that and still adequate for the task of getting around or doing whatever else it is the robot is supposed to be doing. It may not be necessary to have absolute positional information. Relative informational or topological information may be enough to guide local actions and after all that's all that the robot has to do. Obviously, that's not the general case-I said in many cases. The pure navigation problem if we could abstract away from everything about the world except for geometry what you would be left with is a set of problems in computational geometry and obviously there there's a whole hierarchy of complexity that you can into, depending on what aspects of the geometric situation and the motions of other moving bodies and so on you want to get into. Now having said all that, what does AI have to do with this at all? Why is this not just pure computational geometry if we're focusing on navigation as a stand-alone task. Well, the real issue is not so much with pure navigation - in the case where you have a geometric module as it were but with impure navigation. In other words, navigation as a competence of the agent that's inter-leaved with many other competences and many other tasks that the agent is trying to carry out. In general, the system designer and I think Raja said this a little bit in the

previous talk, the system designer is interested in non-geometric aspects of the task as well and he also has a priori information about the task domain. So the problem is not getting from A to B but it's getting from A to perhaps in the vicinity of one of a number of B's that have certain characteristics while maintaining site of the road, avoiding being seen, staying away from poison ivy, making sure that the car goes undamaged and so on. Now all these things are not as easily modelled using the tools of classical mathematics, the mathematics that we use to model space and time. These are not as easily modelled and so we fall back on more symbolic, more qualitative models and the challenge is how to incorporate those into a system that's also doing navigation as one of its competencies, not its exclusive competence. So it's a matter of how to allow these things to interact in a controlled way and get the whole system doing the right thing. Now, there are a lot of strategies for overcoming complexity and I won't go into them all now but I'll make one observation and one plug, one advertisement. The observation is this. While it's desirable to have modular descriptions of competence at the design level, that modularity need not imply that each competence is realized as a separate module at the system level. In other words, there is not necessarily a one-to-one mapping between these competencies, if you will, or these behaviors or abilities and particular modules at the system implementation level. One of the approaches that we've explored and have actually used in programming robots is to use sophisticated compilers that allow the designer to specify these competencies in a constraint oriented way to be able to say things like 'I want you to maintain sight of the road and avoid the poison ivy' and then to have the compiler calculate the interactions between the lower level behavior generators such that those constraints would be simultaneously satisfied in a reactive system. So it pushes a lot of the work onto the compiler and I'll give you a pointer here to a paper called 'Gapps' or about the system called 'Gapps' that was written by Leslie Kaelbling and appeared in last year's AAAI (pg 60 of the proceeding). A final comment on sensor fusion. Combinatorics and robustness say yes to sensor fusion. The idea of having separate sensors - each of which independently controls the behavior is a combinatorial lose in the long run. Because, even if these behaviors control multiple effectors, this is kind of a caricature of Rod's methodology in some ways, even if each of these controls multiple effectors, your left wheels, your right wheels so on and so forth, various things and even if other sensors can be preemptively institute their own preferences to what these behaviours can be by blotting out previous ones. It is clear that you are getting into a linear game when you really want it to be in an exponential gain. In other words, you want the combinatorics of mixing and matching to work in your favour. This is making combinatorial explosion work for you rather than against you. What you want to be able to do is have multiple sensors, or equivalently multiple ?????? information states. Multiple effectors; what will look like fusion here, controlling different ones of the effectors in different ways. What you want is a textured data flow diagram, if you will, rather than a linear one. This argues both for fusion on the sensor side and also for ??????? behaviour specification on the output side. I will not say too much more about that but I urge you to read the 'Gapps' paper and also to read a paper on perceptual updating in this knowledge representation conference that I wrote. Thank you.

WEISBIN

I have only four slides this morning so I hope to be mercifully brief. David wanted something that was provocative and I did not really think there was much that was provocative here so I will try to give you a summary of what I have heard so far and pointers as to where we might look. I will do it in the same style that Dr. Chatila did which is a question and an answer. The first one again had to do with models. To us it seems that what needs to be modelled clearly is task dependent. That was said at least three times this morning and it would be interesting to see if there is any view that says that modelling is task independent. But I do want to point out that models are, can and should be built at different levels of resolution. So, we might be in semantic issue over what is a model? If you are just trying to avoid an obstacle, and do not really care what it is, you might store for example the line or the edge. Someone might call that a model, others might say unless you have a complete scene interpretation it is not. But it is clear that we do have models and they are built at different levels of resolution. Secondly, if you can afford to build models they can certainly be useful. Whether it be in terms of self location, finding out where you are, recognition or global path

planning as opposed to local deviations and to convince yourself that you are not in a loop. Have I ever been to this location before? So, it seems to me that at least one approach was if you have an a priori model, or can determine one then the answer is to use that a priori model and confirm it consistently based on sensory feedback. The second question had to do with the type and synthesis of sensory input. I keep coming back that this is also highly task dependent. For complex tasks, as has been mentioned earlier, our experience has been that no one sensor is usually sufficient. You have problems with resolution, you have problems with systematic interpretation errors. That is not to say that you cannot do simple things with a single sensor. But when you get into very complex operations, multiple sensors seem essential to us. They can be used in a variety of ways: for consistent labelling, for deeper interpretation of the input that you already have from another sensor is another example and to identify complementary features. So we would have said to perform multi sensor fusion where time and resources are available and to utilize independent sensor readings when speed or cost effectiveness dominates. Then there was a notion about whether execution monitoring needed to be preplanned and we do not see at all how you can not preplan execution monitoring. As soon as you pick your choice of sensors in terms of what to plan for measurement, you have already made the determination of what you are going to monitor even before you have begun. Planning and execution demand monitoring of performance against expectation for goal directed tasks. If you are going to monitor execution, you must preplan. Obviously then, the more critical the task the more important it is to preplan systematic execution monitoring. Finally, there was this question about the relationship of route planning to general purpose planning and I did not interpret that to be general purpose route planning. The question is route planning versus planning in general. It seems to us that general route planning is a subset of many or even most of general purpose planning paradigm. Some exceptions might include the planning of dialogues for natural language communications or the planning of joint action such as combined mobility and manipulation. However, this clearly does not seem to be our most important issue and therefore, we would come and say that planning and execution are to be traded off. Planning is only one part of a much broader problem and to urge the panel not to spend too much time debating the issue. Thank you.

BROOKS

Can we have the slides? I have been building a number of robots over the last years. Here is a picture of some of them. They range from about 50 kilograms down to 50 grams and some of them run around on wheels and some have legs, etc. What I want to talk about today is sort of stimulated by my experiences with these robots. Next slide. The topic we are talking about is navigation and I have a problem with talking about navigation as a separate issue. I think when you are building a robot brain, you need to think about how navigation fits in with all the other competences you want the robot to have. You have to see how it fits in with vision, for instance, may be how it fits in with things like manipulation, discourse how these things all come together. Even go so far as to worry about how it fits in with the robot doing philosophy. It seems to me it is a little hard to isolate out navigation from all of these other things. It depends on how the pieces are put together; but, having this view that navigation and vision etc are separate models has lead AI to think about having to build general purpose modules for all of these things. If someone is just working on vision or just working on navigation it tends to get evaluated less. Can your single module solve these particular problems. In general, I do not think there are general purpose navigation, general purpose vision or even general purpose planners. Certainly, if you look at the ethological literature and an neuroethological literature there is not much evidence of these general purpose competences in most animals or even in human beings. So, I think we have a problem steming from the traditional AI view of dividing intelligence up into separate modules which do separate things and the traditional way AI view has led to performance evaluation by asking can this particular module do particular things? I sort of call this puzzleitis. Whenever someone has a planner, people say well can it solve the assessment anomaly. Purposeness emerges from the interactions of multiple modules. We do not have general purpose single modules which are subject to this puzzleitis disease. Instead of having this view of the world, with an intelligence sitting up there where everything feeds through this single intelligence. I prefer a view of the world where we have lots of agents sitting interacting with the world. May be they have some communication between each other and then we do not get into

the questions about fusion and general purpose route planning etc. I want to give an example here. This is one particular robot called Herbert that we have built over the last few years. Herbert has a bunch of different sensors including a laser light stripper which gives a depth map. It has an arm which lets it manipulate things in the world. It has a totally distributed processing system, 24 microprocessors sitting around its belly and it is not a very efficient use of silicon but it illustrates that it is a totally distributed system, there is no central control whatsoever. There are little wires that you plug together to connect processes and send messages between processors and that is how you program the thing. The role in life for Herbert is to collect soda cans. We have finally successfully demonstrated the whole system in June where we switched it on in an unknown location. It wandered around going through doorways going down corridors etc looking for soda cans. It can look for them at a floor level or table top level. When it finds a soda can it reaches out, picks it up and then goes back to where it was switched on and deposits the soda can. It's nickname, by the way, is the collection machine. It is an interesting example because it is a very extreme case. This robot has 50 some parallel agents running which have no communication between themselves whatsoever. All communication is done through the world. Furthermore, they have no state which lasts more than three seconds. Anything that an agent knows about is completely forgotten in three seconds. There is no internal state that lasts very long. I am not advocating that we should build all of our robots this way but we wanted to try and map out a very extreme point in the space of how you could build robots and see what we could get the robot to do. Now, another sort of sensor this robot has is infrared proximity sensors. The point I want to demonstrate here is that with these sorts of sensors it is impossible to ever conceive of building an accurate 3 dimensional reconstruction of the world. For instance, the robot has two rings of sensors an upper ring and a lower ring of 16 sensors each. This was data we actually took from the actual sensors to see how sensitive they were and which direction they were pointing and after we had been running this robot for about two years successfully navigating around assuming that the sensors were very uniform in what they were doing it turned out that they were totally non-uniform. Some of them did not even work and we had not even noticed that. The very idea of building a three dimensional world model from this sort of sensor is totally ludicrous. It turned out that we did not need to characterize exactly what the sensors where telling us. We work in a piece of design space where that did not matter because we were not trying to construct 3 dimensional reality. I noticed that a lot of other panelists this morning are sort of backing away, well, except Raja he is still got this Cartesian envy and wants to stick with the 3 dimensional reconstruction of the world. In general, I think that people are backing away from that motion. This is another sense that the robot has on board a compass, a magnetic compass and this is some data taken around the lab. These little circles have a dark bar which indicates which direction the compass thinks is north at this location and then west, east and south and you can see this compass is a magnetic flux gate compass. You can see it is not very good. North, south, east and west are not orthogonal in lots of locations, some they are. The direction of north varies a lot. So, trying to use such a sensor to build an accurate model and relate coordinate systems accurately is again not a very good idea and can lead you to lots of engineering and using expensive sensors where it turns out you do not need them. This is part of the control system for the robot or what you could call navigation. If you really want to label part of the robot as doing navigation. It is a bunch of different agents which are fairly simple finite state machines which get inputs from various sensors. Here is a couple from shaft encoders. This is from an infrared proximity sensor map, compass, separation of the fingers is an important one there. Now, what is happening here, and this a little like the diagram Stan drew by hand at the end there, is we have multiple sensors coming in; they are not getting fused into a central model, they may be getting fused into partial models locally within these behaviour generating modules and then the outputs of these come through some sort of priority network and eventually get to control the actuators. It is interesting, there is one cute thing here, I said the robot is able to find its way back to where it was switched on but it has no state, it has no memory, well how can it do that? Well it looks at whether its fingers are open or not. If its fingers are open, it means it is not grasping a soda can, if it is closed it is grasping a soda can and it uses a different set of navigation rules essentially and so when it does not have a soda can it reacts to the environment in one way, when it does have a soda can it reacts in a different way and it is carefully designed so that it is able to retrace its steps without having any memory inside its head of what the world was like. It uses the world. Now, with all of these parallel processes running

trying to drive the actuators one might think you get a lot of randomness. It turns out that from collecting real data we actually get in a given situation fairly consistent behaviour; trying to navigate along a corridor and this is from 10 successive rounds and it is rather consistent. Let me skip over these and quickly answer the five questions. First one was what aspects of the robots, environment need to be modelled internally by the robot? I would answer in some cases that the extreme you do not need to model anything at all. You can use the world directly. Other cases, you do need to may be model the environment in terms of what the robots capabilities are. Not in terms of geometric reality but in terms of what the robot can do such as follow corridors, go through doorways. Rarely do you need to describe the world in terms of objects and their relationships to each other but all of these things can co-exist. What kind of sensors? Again, I agree totally task dependent. It is hard to imagine when you need really general purpose vision and I do not think that even humans have general purpose vision but may be Raja can push for it if he really wants to. Fusion or fission, I think it is important to realize that you do not need to build general consistent world models. If you do think that you have to fuse everything together into one central model, it makes you think you need better sensors than you do. Do you need execution monitoring? I think the idea of having a planning system and a separate execution monitor to see whether the plan is carried out is just a completely wrong way of slicing up the problem. We need multiple agents which individually are doing their own execution monitoring in a way that is actually operated. Each of them knows what it is supposed to do and so execution monitoring is intrinsically built in. It should not be separated ?????????. And lastly, is general purpose route planning any easier to target than general purpose planning? No, I do not think that general purpose planning is an easy target or even a doable target. I do not think that general purpose route planning is at all useful. You do not need to plan routes in general you need to carry out tasks and it is just having to split the problem up into modules and separating those out that makes you think you need to do general purpose route planning. You need to get to where you need to be; not to plan a route to get there. In general, it turns out that now existing robots are even close in performance the way you can use a general purpose route planner. They do not go more than a few yards before things go wrong. It is not really an issue. Thank you.

THORPE

The rumor was that ever since Rod got tenure he got mellow and I am glad to see that is not the case. I want to talk about what real robots do. A large amount of what we have done in the community is to build robots to show off one particular theory. If we want to have the theory that robots ought to be built out of subsumption architectures. It is the architectures that drive what we are doing and not the fact that we are really trying to accomplish something. If we are really out to accomplish some real task then we have to hit the hard questions about do robots really use AI at all. What kinds of things are they going to monitor? What kinds of things are they going to think about? What level are they going to be doing their model at? What do I mean by a real robot? First of all, it has to work in some sort of unconstrained environment. I put "outdoors" in quotes because I do not want to rule out an unconstrained indoor environment; but, I do want to rule out environments where we have carefully painted markings, carefully set up so that we can do things easily. I prefer environments that are harder to model. I prefer environments which are less structured. Secondly, real robots are a significant enough size that they can do significant things. It is interesting to build little things that run around the lab; but, if you are going to do a task like explore the surface of Mars, the robot has to be big enough to carry scientific sensors, it has to be big enough to pick up a rock, it has to be big enough to step over the wide crevices etc. I am talking about a large enough robot that you do have to worry about issues of size. Most importantly, I am talking about robots that have real tasks to do. My model for a prototype real robot is the Navlab that we have been playing with at CMU. From outside it looks like a mild mannered Chevy Van but on the inside it looks like your favourite computer room. You put the computers in there, we put the generators in there, we put the sensors in there and we put the researchers right there on board and the graduate students ride much more code when they are travelling with the NAVLAB. This is just our test bed. The real robots that are the spin-offs of NAVLAB research look like this. This is a real robot that its mission in life is hauling ore or coal out of the strip mine and when you build something like this

you want to be very careful about getting it right the first time. It's notion of an obstacle, by the way, when it is doing obstacle avoidance and detection is about the size of this table which makes some problems easier. Here is another idea of a real robot. This is one that is under construction at CMU. For generic planetary exploration where the generic planet has characteristics very similar to Mars, it is about four and one half meters tall to the top and has six legs. The legs work in a weird pattern where they move back through the middle of the body so that it is called the leg beater. Well, we are at an AI conference. Do real robots really do AI? Well, they do not play chess and they do not talk and they do not play with blocks so it is real easy to say therefore they do not do AI because they do not do any of the things you would have seen in old IJCAI proceedings. A little bit more seriously they do not do a lot of symbol pushing in the traditional AI sense. They do not look like expert systems, they do not do induction and deduction and backward chaining and forward chaining. They do push symbols; but, the kinds of symbols that real robots push are geometric symbols. The kinds of tasks which are the most difficult for real robots are sensing the world and then dealing with the geometry of what is out there. A real robot will do filtering and estimation and will use maps. The maps need not be globally accurate unless you are sailing across the ocean. They do need to be relatively accurate. The easiest way to get relative accuracy is to buy more expensive sensors. The next easiest way is to use clever filtering. So, all of the statistics literature and common filtering that allows you to with very little extra computation get a lot more bits out of your sensors. That kind of stuff is very important. Maps are vitally important. Whether they are built in advance or whether they are built during the run. If you have an accurate map that tells you where to look that tells you what to expect it vastly increases the ease of doing your job. The fusion and fission question is bogus because of course you have to do both. If you have a very good sensor there are several modules that want to use it. If you have lots of sensors they may need to be all fused into a single representation. In fact, robots often use lots of different sensors and the lemma is that cameras are cheap. By that I mean, it is easier to put a camera pointing at the road and a camera pointing at the landmark and a camera pointing behind then it is to hire some clever graduate student and have him build the program that decides when the single camera ought to be looking back and forth and trying to do all of the coordination. It is easier just to bolt on lots of cameras. Real robots have to communicate with real people. The communication may just be in terms of telling them what their job is to do and you may do that just by building them cleverly. But, in any case, whether you are building them or whether you are programming them, whether you are talking to them, you need to have some easy way of telling them what they are supposed to do before you kick them loose and let them go. Last comment, if I have the choice between more computers and more capable hardware for doing a cross-country mission I will take the big wheels any time and leave the computers off. As long as the people who are designing the hardware talk to the people who are designing the controller and build it as a unified system that you can actually control. How do real robots plan? Well, I have not found real robots yet that are travelling salesmen. I have not found real robots yet that are moving pianos. So, I do not think that the optimization questions in planning are that important. I think that building plans which are sufficient, which suffice, are much more important, therefore, the path planning is pretty easy, in general, therefore you can do it fast. If you do it fast you can keep path planning every time both on a local scale and on a global scale so every time you've moved every time you have new sensor information you do not have to worry about monitoring your plan because you keep building new plans and checking your plans as you go. You may lose if the world is serendipitous. If your robot bank robber has planned how to go into the bank and he sees the money in front of him that fell out of the brinks truck and he considers that as an obstacle and goes around it. That does not happen in the real world. The real world, usually its tougher than you want. It is not easier than you want so you do not have to keep monitoring your plan to see if by accident you have happened to have picked up the money you were looking for. So, what do real robots have to model in order to do real things? Well again, the obvious answer is there is lots of different things and lots of different levels. You may see an object which for some purposes just an obstacle for another purpose is a stop sign. How you have to worry about the semantics of stop signs and you have to worry about if it is written differently in Canada than in the US and so forth. Or it may be the stop sign at the corner of whatever which then has to tie back into the map. Consider for example what you would do yourself if you were driving down the road and you saw a moving object. You may make plans to avoid it.

That may be all that you have to do to build a sort of a slow pokey robot that pokes down the road. If it is a blue Chevy then you may know something more about what its behaviour is going to be. You may have modelled it a little bit better. You may be able to make better predictions about what the blue Chevy is going to do. If it is a blue Chevy with somebody driving it slowly in the left-hand lane with the blinker on then you have a completely different model in your head if you are a good driver of what this person is likely to do. You will know how much wider of a margin you ought to leave than the typical blue Chevy and you will react to it completely differently. If you are in Pennisylvania and you see a blue Chevy with we serve and protect written on the side and a gumball machine on top you hit the breaks. If it is your brother, you may pull up behind him and honk or something like that. So, you may end up when you really build capable robots and when you really are out there interacting with people, you may end up having to do an awful lot of AI in a much more tradition AI sense in order to do this kind of modelling of people's behaviour in order to decide how you ought to drive in their midst. Let me give you a couple of pictures of the things that we have modelled and of what is important about them. Hebert and Quan have been modelling the errors in vehicle motion and the errors in range data so that they can build accurate maps. This map is a cross-country terrain map built of 120 different laser scans. The better model you have of the accuracy of each laser scan the more accurately you can put together this big map which is useful for navigation and useful for science. It is important to model all of the kinds of characteristics of what the vehicle can and cannot do. These are some of the traditional ones if you get the vehicle tilted too far it flips over etc., but there is a lot more than that that you can model about a robot. Tony Stence's path planner includes things like how accurate is the vehicle motion. You could see may be two obstacles that you know that you could get through but you know by the time you drive up to there your accuracy is so poor you are going to ram him. So, you have to plan a path that gets up close, looks and then sneaks through. That kind of modelling of the vehicle is important including the non holonomic constraints so that you can do paths with several different laser scans and slide the vehicle out and around obstacles. Another set of things to model. If you can build a map, then on a subsequent pass you could do less perception because you can predict where the road is, you can predict where the landmarks, in this case, where the trees are. You can drive either more quickly or more accurately or with fewer sensing. The last examples what Carl ???? is working on for his thesis. If you can model what is on a typical road you can predict how to look for the different features on the road and you can predict what kinds of failures you are likely to have when you hit weird conditions like intersections, like shadows. The better you can model the lighting, the better you can model your cameras, the better you can model the performance of your operators, the better you can model the shape and the appearance of the road, the better job you can do navigating. So, while you do not have to have perfect models for everything the better models that you have the better a job your robot can do being a real robot doing a real task. Thank you. Okay.

MILLER

For the next couple of minutes I just want to go over a few observations I have made during people's talks and suggest a couple of things to perhaps spark a little discussion. As soon as I am finished we are going to talk among ourselves for a few minutes then I would like to encourage questions from the floor. First of all, just to summarize a little bit, some of the important things that I saw various people say. Scott Harmon brought up the issue of non-geometric hazard detection and for those of you who have not thought about that before let me give you an example from a Mars Rover domain. On Mars, and on Earth, there is this kind of a lava surface that is a gas filled bubble of rock, very thin shell and it is possible if you had a vehicle you would look at that and you would see a small rise up ahead but if you start to go over it you could crack through the shell and fall in. That would be generally considered a bad thing. Also on Mars, there are probably dust filled craters which are shallow craters filled with this low load bearing strength dust and it is possible that a vehicle could drive into that, it would see a flat surface ahead drive into it and become mixed. So, you have to be able to watch out for things like that and I think that is an important point. Landmark navigation is something else he brought up. Experience, there is not a whole lot of experience on mobile robots yet and that is definitely something that needs to be built up and I thought that the other insightful thing was suggesting not to go out to eat with Rod Brooks, at least not if Rod is buying. From a

difference point of view, Raja brought out that natural terrain needs vision plus other sensors. You suggested colour, some sort of range data to build up an accurate map. Also, there is this question of to sort of be able to plan out your execution monitoring you need to know what kind of unexpected things are going to happen. That is a real difficult issue that I personally think to get a good handle on; but, something that ought to be discussed. There is also the question that I had that came during the discussion is one of the issues of natural terrain is that there is not well defined obstacles. There are not walls set up like there are in indoor domains. There are hills and rocks and things like that and it is possible a rock is non obstacle but if you put two rocks and you space them just the right way, then you can have a obstacle. As an example, this is one of the robots we are building at JPL. It has six wheels. It might have briefly eluded too in a couple of the talks. It can go over a half meter rock very easily. But if you put a half meter rock here and a half meter rock there and a half meter rock there it is not going to go anywhere at all. It is not just individual obstacles but it is the whole context of the terrain that is an issue. Some of the points from Stan Rosenschein's talk is that relative navigation is adequate for many tasks. That you do not need a model and that perhaps is a controversial statement. Also, that the real difficulty in robot navigation is doing robot navigation and all of the other things that you want to do. Not just getting from A to B but getting from A to B in one piece or in time or with the bagels still intact or whatever you need to do. Chuck brought up the question of can models be task independent and the answer is probably no. You really do want to set your modelling and your sensors and everything very task dependent. He also brought up the idea of using multiple complementary sensors and I was kind of surprised that nobody actually explicitly said the idea of diagnosis. If you have all of these different sensor modelaties. Whether you are fissing them or fusing them, when one of them starts to screw up it is nice to have other sensors so you can compare them to and try and figure out what is actually going wrong. Rod mentioned that evaluation and design should be very task specific again and really stress the idea that accuracy is not necessary for many problems and that is something that has not really been discussed a lot just once again, going from the space domain if you walk down the hall you can see the Voyager you can see some of the pictures coming from the Voyager space craft which is a robot that we launched about 12 years ago going around Neptune. Occasionally accuracy is very dependent you just want to be able to avoid collisions in some cases. While this is somewhat of an exaggeration on things like Mars Rover or if you are in nuclear reactor or doing anything sometimes accuracy is very critical though often it can be avoided. A couple of important comments from Chuck Thorpe real robots go from A to B, I think is my attempt to summarize what you were saying. It really is; there are some things where you want; it is important what you want the robot to do, not necessarily how it does it. Also, an interesting thing was the idea that not only are there geometric and non-geometric obstacles but there are sematic obstacles which I think when you start dealing in more human domains I think that is important. So just a couple of additional questions I wanted to throw out. These sort of tie into the idea of where this fits in with AI-if it does. The whole idea of robot planning is goal directed planning since it is obvious that robots have to have reflexes somehow. How does that effect the idea of goal directed planning? What is the real goal that you are trying to give the robot? For a generic vision system or planning system, once again, what should such a system do? There are lots of people who have been working on vision for a long time and they produce lots of huge systems that take in data and process it and produce other data that is interesting and when you look at it it is obvious of what it is but there are very few robots that use any of that and the question is why. Is that because it is just an integration problem or because it is not producing the kind of output that is actually useful for a real robot? That is one of the reasons why I thought this panel was important is that real robots force you to integrate and to face somewhat of the real world. Just reflecting on Chuck's comment: real robots do not prove theories. Sometimes they might. It depends on what you are trying to get your robot to do. So, one of the main questions I would like to throw open is what is a good problem domain? Most of the domains we have talked about that have been mentioned the NAVLAB and the Mars Rover and even the coke can collector it is basically go from A to B and get something and drop something off. Or, go from A to wherever and back to A and drop something off. Nobody mentioned the helpful household robot that has to figure out when it is making the bed whether it has made the cat in the bed or whatever. Things like that. Problems that might interact a little more with some other problems but is the reason why nobody has mentioned because it is just too hard or cannot be done.

I am throwing this open to the panel and let us have an argument for a few minutes.

BROOKS

Can I do something else? I want to summarize everyone's talks. You want to summarize everyone?

I think all of the panel members are suffering from different diseases. I just put them up here so you can see explicitly what they are. Scott is afraid of complexity. He is afraid if he has got this non holonomic robot he will not be able to figure out how to control it. Raja with his French background, of course, has his cartesian insecurity where he really wants to know everything about the world and where it is. Stan sort of surpressed it today; but, he really has this logicitis disease where he wants to be able to logically prove what his robots can do. Chuck does not want to deal with academic theories he wants his robots to do real stuff. He is not worried about these academic questions that we in our ivory towers deal with. I am always interested in showing traditional AI is no good. Lastly, Chuck has this envy he wants a Winebago for his robot. He does not want a little toy. I should point out that with my own disease from the acronym you can see why no one wants to go to dinner with me.

MILLER

Somebody please say something.

WEISBIN

In my case, I heard two or three new things from Rod Brooks that I have never heard before. One was a word that says in some cases. Another was a word that said task dependent and there was another word that says sometimes. So, we are moving somewhat from absolutism to a more compromising approach. Let me give one example of a very real problem which may be somewhat down the road but is very real. Oak Ridge closed its gaseous diffusion plant some time ago and it now has to be decontaminated and decommissioned. There is a lot of alpha activity all in the pipes and they do not want people there. There is 230,000 tons of metal that has to be navigated to, cut, washed, moved. They are very interested in: can robotics contribute to solving that problem? I go back to the point that several people here have made. Navigation is not the total robot problem. The problem is to do something useful, which goes both beyond collecting soda cans and what we do in our laboratory: press little plastic buttons too. It works both ways. But certainly tasks that are hazardous to human beings unstructured and require real work are things that are ideal for the robotics challenge. Are we ready and do we want to face that kind of thing?

MILLER

I guess the answer is no. Any response.

BROOKS

Are you offering funding?

WEISBIN

There is about $13 million next year for this activity.

BROOKS

It sounds like a great idea.

CHATILA

Usually in these discussions, we sent around Rod's theories so this one will not be an official case. I want to ask you Rod what is a soda can.

BROOKS

A soda can?

CHATILA

Yes. I mean what features.

BROOKS

The robot collects soda can like objects. Soda-can-like. They have some geometric properties so there is some object recognition of that level.

CHATILA

So it has a model of a soda can?

BROOKS

It has a model built in. It is not trying to merge form all its senses into that single world model and that is where I go along more with Stan that you do not have this one central repository. Now Chuck, is moving towards that goal with his multiple scale models and I agree wholeheartedly with that. I do not have any problem with that.

ROSENSCHEIN

Now, let me say why I am hung up with logic and sematics. If you wanted your robot to collect more than soda cans; but, you wanted it to be able to collect soda cans or cups or other things, imagine if you wanted to be able to task your robot at run time a bit by giving it goals like bring Leslie the bagel and so on and you did not want to have to re-program it to be able to say bring Jim the cup and bring so and so the such and such. In other words, you would want to get into a ???????? analysis of what the sematics of the various tasks where they were carrying out and also what the different conditions were that it could recognize and once you start doing that, before you know it you have kind of tripped into the semantic analysis associated with more traditional approaches. How would you have to change your robot in order to be able to get it to move things from place to place that were not soda can like but may be they were big and bulky. Obviously, some of the navigational stuff still applies; but, a lot of the low level triggers and cues may be quite different.

BROOKS

Well, that is a fair question but I think it is a little unfair to ask it. See, you use your example of bring Leslie the bagel; but, the only way ????????? could bring Leslie the bagel was because you told it there was a bagel on your head. If you had told it that was a cup on its head it would have just been the same process.

ROSENSCHEIN

Clearly, the reason why telling it that compensated for its lack of sensory abilities. In the future we would like it to have better sensory abilities. There are things it might have known about but it could have done on its own. I do not understand... There is still the issue of how configurable is

the robot at run time? How many different kinds of tasks and how much flexibility can you expect it to exhibit?

BROOKS

There is the real question of how, from one point of view, that is something that we may need to address in the long term but we can ask how flexible do we need our robots at run time to be useful still and the analogy I like to use is: these are really useful devices but we never tell them what to do. We put their hide down where the flowers are. They go and do their stuff. There is a large class of tasks.

ROSENSCHEIN

One final comment. There is obviously a spectum from completely hard wired industrial style nonsensor based robotics all the way to fully intelligent, you know, robustly reactive robots and so on. It seems to me that you have moved a little bit towards the direction of flexibility.

BROOKS

The methodology on the adopting is that we do not have enough constraints to go all the way so let us work on this and find out what the real constraints are.

MILLIER ?

Stan your video upstairs starts out with saying: Suppose Stan was replaced by a robot and I assume from that you are interested in having reconfigurable robots that can do lots of different tasks at any one time. Rod I assume you are much more interested in saying suppose this cockroach was replaced by a robot. I have seen cockroaches delivering bagels. Chuck you have been very quiet.

THORPE

Let me put up a slide here. There was a NATO advance study workshop of a couple of years ago that most of the people on this panel or their co-workers attended and at the end of it I asked people for their guesses as to when some of these technologies would mature as to what was going to be the pacing factor that was going to make each of these technologies more difficult than the previous one. You can see some of the guesses there. Part of what is going is on is more and more difficult modelling. If you are just wandering around you do not need much modelling; but, I think that if I am correct that Rod's robots do not guarantee coverage. You will find most of the coke cans in most of the places but in order to be able to get back home without a map it has to have some heuristic like only left turns going out and all right turns coming back which keeps you from finding some of the interesting offices that may be where they store their stock of new coke cans or something. Is that true Rod?

BROOKS

Can you hear me. I do want to point out that I am not saying that that extreme position with no internal state solves the problem. That really was meant as an extreme demonstration. You are again observing puzzleitis. But can it do X? Well it was not designed to do X. Just like your real robots do not play chess. No one ever asks about a chess program but can it climb mountains. This game we play about what we expect certain things to be able to do and we have certain sets of expectations but I am not sure those sets of expectations are always a valid set of expectations of what a robot should be able to do and in general a robot that cannot do everything can still be useful. If you look at any human worker, they do not do everything by any means and a good manager compensates with workers that are able to do different things and do different tasks well and not well. This quest for perfection all of the time and it is just in the world models and in the behaviour of the robot leads

us to try and build these super robots. Your solution is to build a big one which can stomp on anything to be perfect. I think that is the wrong view point.

THORPE

Part of what happens is that these have to be more and more perfect as you go down. Cleaning the floors it is nice if it is pretty good. A wheel chair you can see that nobody wanted to guess how long it would be before we had them because they have to be perfect or else the lawyers are on top of you. If you get down to something like a street sweeper that has to work out in real traffic then that has to have better models and it has to be a lot closer to perfect. The exceptions are if you want to do something like being a chauffeur, Professor ????????? says no problem. Seven years we will have it. I think that is an outlier. I think the better numbers to look at are things like the industrial cleaning number up there, where the reasons for the variances that Jim Crowley was not building a robot to do industrial cleaning and therefore did not have to worry about modelling issues and Raja Chatila was building a robot to do industrial cleaning and did have to worry about those issues.

BROOKS

Raja wanted to worry about the modelling issues he did not have to worry about the modelling issues.

CHATILA

We still do not have a general cleaning robot do we? We do not. Especially for example if you want to take.

BROOKS

Do you want a general cleaning robot?

CHATILA

Well I do want a robot that cleans and plays while people are in it. We haven't achieved this yet, Right? By the way, my Cartesian insecurity I believe is also a very important issue for the robot not only for me. Because the robot does not want to hit things but like Chuck pointed out it wants to go to a given place more or less precisely to do something. I want to just show two slides. I want the robot to know, to realize what it is looking at. What is this? Can you tell me what this is? You need for this several sensors. You need vision and something else. For example, when you use stereo vision by itself that is what you get. Edges, unlinked you do not know where the matter is you do not know what/where free space is exactly. This is projection underground. You do not know the exact shape of objects. When you use a very common device, 'a horizontally scanning laser range finder', this is what you see. You see edges if you care to model the points, the measuring points that you obtain. You do not know about the third dimension. Well, sensory data fusion provides you with a more complete representation. You can link edges where they are not linked by stereo vision and you can deduce plannar facets representing the environment. That is useful representation for the robot for navigation and for space understanding. You need the robot to do tasks. Space understanding? Well, suppose I want to go to the panel. There is something here. I want to go to this place. How to describe it? Space understanding is having internal representations, descriptions of the environmental objects because I want to go to these objects. I want to do some tasks related to these objects. Tasks. I do not just want to go randomly to pick up something that I will identify coursely because I can as well pick up a soda can as any cylindrical shaped object even if it is not a soda can. I do not want the robot to touch it whatsoever. I want the robot to go to a given place to achieve a given task for example to a dock, to a workstation; but, while doing this I want it to avoid obstacles, of course, to plan a safe path, to also execute a path as optimally as possible because I have limited energy, I have limited capacities so the path should be, for example, in some cases not

only go to a given place, plan a free path but in some cases I would use other sensors. For example I will use wall following because in this area I will notice that this mode is sufficient, therefore, I need several sensors not only for fusion but also for different tasks and I need to give to my robot a complex task not just build a robot for a given task. If I change a task, well, I will build another one. No. I want a single robot, reconfigurable that I can program and that can adapt to its environmental conditions.

HARMON

I would like to take Chuck's slide for a minute here. This issue right here. The liability issue is not because we wanted to make that robot perfect it is because nobody at the meeting could figure out how to overcome the liability issue. It was not an issue of being able to actually make the robot drive a paraplegic person around we all felt that was something that we could actually do. But, if something went wrong, there was some kind of failure or something like that we did not know how we would handle the liability issue.

????????

You might also want to take this opportunity to defend your name next to the tax simulator. That was not something that you believe but that was something you said that the navy believed.

HARMON

I think that is probably still reasonable. There is a program going on now to do that. From these talks, you might think that we have spent all of our time making definitions and arguing about definitions as opposed to building robots and I have put some words that I heard over and over again. One that I do not know quantitative fusion. That is a new one to me. I will have to ask Raja about that. Rod talking about general purpose modules and then the whole issue of real robots and what these terms serve to do is put us in groups. See Rod is against modularization in some cases but he modularizes people and so we are all busy trying to make arguments here and the way that we do that is we classify people in particular groups and then they respond to it and get mad and we get up here and talk. Models do not necessarily exist in one place as you saw from the discussion between Raja and Rod. Fusion likewise can be done in many places but I have a question of what is real fusion. Obviously, it is not what the sun does now anymore since we have seen perhaps that it is being done in a jar. Sensors can serve many purposes and many servers so that is the reason for making them more broadly useful. Notice how I stayed away from the term general purpose. My question is if you have an expensive sensor like a camera serve a number of users of that information then is that making it general purpose. I do not think that is necessarily true. In response to Stan, I very much admire the work he is doing and I would love to have something that does rigorous analysis of robots and robot systems but today I think everyone of us here, with perhaps the exception of Stan, is an artist. That is to say we build a little and go out and play with it and see if it is exactly what we want sort of like a Michelangelo carving the Pieta. I would like to finish by saying that you may be all deterred from hearing all of this ranting and raving and arguing amongst us but perhaps the only thing that we can do is encourage you to actually get involved in this. Mobile robots are pretty accessible, a lot more accessible in fact than manipulators and so I suggest to anybody who is fearless enough to build one and buy as much as you can so you do not get stuck in building little pieces that you do not need to. Although Chuck and I disagree about this. Do not try to convert a human driven vehicle. It brings its own problems that you do not necessarily want to solve. Plan for more computing than you expect. Build lots of extra space and power for that because it will probably take it. Even if you wanted to do something simple. You saw Rod's simple robot and all of the wires and computers hanging off of that. Use as many sensors as you can with overlapping coverage because as we said a single sensor is not reliable and yes I do have a fear of complexity only because I want to try to get something done and if you do not keep it simple, you are going to be overwhelmed by the complexity and you will not achieve it and I think that Rod is a perfect example of making it simple and actually getting a product out. Thank you very much.

MILLER

I think that we are pretty much out of time. I would just like to thank all of the panelists for their talking and to you for listening and I think we have time for one or two quick questions.

QUESTION:

Do you want your robots to learn? If so, what and are we ready for it now?

MILLER

I think I can answer for the panel by saying. Yes, everything and no. Next.

QUESTION

Hello, I am Ken Laws? from the National Science Foundation, I have the robotics and machine intelligence program. First an announcement if anyone would like to come and talk with me and get some help on submitting a proposal I would be glad to help. I am even more interested in talking with you if you would like to replace me in this position because I am scheduled to leave in a year. Now for a question. What should be the role of the NSF in funding robotics research? Should I be funding theoreticians who come fairly cheap? Should I be funding people who already have robotic test beds because they can experiment with real world data or should I be buying these cheap commercial platforms for every academic who wants to build another hall walker?

MILLER

Yes. Why are you asking us?

LAWS ?

With a total budget of 1.5 million for the entire country. Do not fund the theorists.

I think there are probably six or seven labs here which are probably most in need of the funding.

BROOKS

Seriously, I think you were a little disparaging at the end there. Should you fund people who do not have robots to get some and build yet another hall follower I do not think that is a bad thing. I think the cost of systems that can be bought is such that with 1.5 million you could find a number of new places. If you funded places which already had an AI component or a vision component it would give them new sources of data which would I think have very positive influences on the work they did when they weren't just working off a set of tinned canned images that they had on disk and I think that could be very beneficial to the field as a whole.

????

There are some attempts afoot to build standardized robots, standardized controllers and even standardized sensors that may provide some of the pieces cheaply for groups that want to do work on others of the pieces. Some of those attempts are ill-considered but their heart is in the right place.

????

The more people we can get involved the more ideas and the more vibrant the field is going to be

as a whole and I think more productive.

I think we have time for one more question. Over there.

QUESTION

On the development ???????????? of tasks seems to come before or flexibility. To what extent do you feel examples of real mobile life should provide guidelines to the evolution of mobile robotics?

BROOKS

I found looking at real mobile life to be very inspirational. It has highlighted for me some of the things that you do not need to do in order to have mobility. When you look at insects for instance. Chuck was very upset about me building small robots. He thought only big robots were important but I think small robots and small insects have lots of advantages in their strength to weight ratio which is very good and so they can afford to get by without lots of cognitive ability so we can get useful robots by building small ones earlier than we could by building great big ones unless they are really big and they can just stomp on everything. I think looking at the ethological literature and seeing how real and also simple animals achieve their daily existence with fairly simple processing can lead to lots of simple solutions but what otherwise might appear to be hard problems.

????

They are also very durable. You can take insects and smash them into objects at very high speeds and drop them from high heights. I think we should be doing that with our robots too.

They are very cheap relatively.

I want to thank everybody on the panel and thank you for coming.